Method and arrangement for instruction word generation in the driving of functional units in a processor

The invention relates to a method for the generation of instruction words for driving functional units in a processor, the instruction words comprising a plurality of instruction words parts. Each instruction word part respectively drives a functional unit. In this case, a sequence of primary instruction words which originates from a translation of a program code undergoes fractionation into program words before a program sequence. During the program sequence, under the control of a program word which has an information part at least of the width of an instruction word part, an instruction word is taken from a row - determined by a reading row number - of an instruction word memory that can be written to row by row and is altered by means of substitution of an instruction word part by the information part of the respective program word. It is then written back to a row of the instruction word memory, the said row being determined by a writing row number. After generation - effected in this way - of an instruction word corresponding to the primary instruction word to be executed, the said instruction word is output for driving the functional units.

The invention furthermore relates to an arrangement for carrying out the method mentioned above.

The German Patent Specification DE 198 59 389 C1 describes a method of the type mentioned in the introduction and an arrangement for driving functional units in a processor.

In this known solution, the program word contains, in addition to the information part, at least also the information about the writing and reading row numbers. This necessitates a width of the program words which, on the one hand, with the requisite processing and

decoding of the control information, produce limitations in the processing speed of the task-related data. On the other hand, the high data width during the program word processing necessitates high outlay on hardware in the realisation of the processor.

It is an object of the invention to reduce the data width of the program word to be processed in order that the outlay on hardware and thus the costs for the realisation of the processor are kept low.

This object is achieved in respect of the method by means of the characterizing features of Claim 1 and in respect of the arrangement by means of the characterizing features of Claims 8 and 9 with the corresponding features of the prior art.

What is realised in this case is that the required reading/writing control information is provided by a read pointer register, which has stored the reading row number, and by a write pointer register, which has stored the writing row number, per program word processing.

It is expedient that the largest possible number of instruction words that are to be executed are successively compiled and stored in the instruction word memory, so that they are combined in blocks. Thus, it is possible to set the read pointer register and the write pointer register with the number of successive reading and writing row numbers in a ring counting mode. This number is provided by the content of a block length register.

In a particular refinement of the method according to the invention, it is provided that the program word has a set bit by which, given a set active state of the set bit, the instruction word generation is interrupted and the register contents of the read pointer register,

and/or of the write pointer register and/or of the block length register are set by the content of the information part of the program word and, given a set inactive state of the set bit, the generation of the instruction word is carried out.

What is realised with this solution is that the required information for the read pointer register and the write pointer register and the block length register for the execution of a further block of instruction words is identified, decoded and, not like a program word, processed further. As a result, these registers can be preset.

What can thus be achieved is that in each case the row of the instruction word memory for the generation of the instruction word corresponding to the primary instruction word is addressed by the program word sequence which contains the instruction word which must undergo the least changes.

Writing that is more favourable for the further instruction word sequence can also be chosen. By way of example, if a plurality of program words are required for the generation of the instruction word corresponding to the primary instruction word, that is to say a plurality of intermediate steps are required, it may be expedient, in the case of each intermediate step, to read from the same row and to write to the same row again. The read pointer register and the write pointer register are thus stopped, which can be determined by the register settings.

Afterwards, it may be expedient, with a specific repetition rate, to allow the writing row and reading row numbers to descend or ascend, beginning from a specific number in each case. That, too, can be realised with a renewed setting. In a supplementary refinement of the method according to the invention, it

is provided that in the case of a program word with a set active state of the set bit, the content of the information part is stored in the read pointer register, the write pointer register and block length register.

An advantageous embodiment of the method according to the invention provides for the instruction word memory to be divided into a first instruction word memory page and into a second instruction word memory page each having the same row numbering, and for the synthesis of the instruction word, the instruction word memory page to be called to be determined by the content of a page register.

In this refinement, the different groupings of the instruction word memory, the instruction word memory pages, are addressed by the content of the page register, in these instruction word memory pages the row addressing being performed with synchronous read register pointer and write register pointer.

Thus, instruction words are set up blockwise in the respective instruction word memory pages and executed, the selection of which is performed only by the indication of the page memory content provided with a smaller bit width and not by means of the addressing of the start values of read pointer and write pointer, for which larger bit widths are necessary.

The object according to the invention is also achieved by means of the features of Claim 5. It thus becomes possible, even in the case of a configuration in which the writing and reading row numbers is [sic] contained in the program words, to achieve a reduction in the width of the program words, since, in particular, in the case of a large number of rows in the instruction word memory, page division enables the row numbers to

be kept smaller and thus the corresponding information in the program word to be kept narrower.

In a further advantageous embodiment of the methods according to the invention, it is provided that an interrupt signal immediately triggers, at the processor, during a processed first task, on an instruction word memory page buffer-storage of a left processing state of the first task on a global memory and then the execution of a second task on the unprocessed first instruction word memory memory [sic] page or the second instruction word memory page, and that, after the ending of the second task, after restorage from the global memory, the first task is continued in a manner rejoining the left processing state of the said first task.

This solution according to the invention ensures that tasks of higher priority, which the processor must execute, said tasks having to be executed quickly, is [sic] advantageously incorporated into the entire execution of the program word processing without a time delay and with an extremely low outlay on buffer-storage and addressing.

In a further refinement of the methods according to the invention, it is provided that a prefetch unit controls the set-up of the instruction word memory, and that, independently of the processing state of the current task, the prefetch unit provides an additional instruction word in an unused row of the instruction word memory or in an additional instruction word memory, if no new instruction word is obtained during the execution of a current task.

This solution ensures that, during the execution of the instruction words, delays in the provision of new instruction words are minimized in that, in waiting times for storing a new instruction word in the

instruction word memory, this time is used by the prefetch unit in order to process a program word that is subsequently present for processing and to provide a newly generated instruction word in a separate area of the instruction word memory for instruction word processing.

One refinement of the arrangement according to the invention provides for the instruction word memory to be assigned a generation unit.

In a further refinement of the arrangement according to the invention, it is provided that a block length register, a read pointer register and a write pointer register are arranged in the generation unit, the read pointer register being assigned a read pointer up/down counter and the write pointer register being assigned a write pointer up/down counter, whose modulo counting properties correspond to the content of the block length register.

The invention will be explained in more detail below using an exemplary embodiment. In the drawing:

Figure 1 shows functional blocks and sequence of the method
Figure 2 shows a block diagram of the generation unit

In the case of the method - shown in Figure 1 - for instruction word generation in the driving of functional units 12 on a processor 13 a sequence of primary instruction words 2 comprising a plurality of instruction word parts 4 is generated in a known manner from a program code 1 by means of translation before a program sequence. Furthermore, the sequence of primary instruction words 2 is fractionated in the program word generation 8 and stored as a sequence of associated program words 17 in a sequence memory 9.

In a program sequence, before outputting from the sequence memory 9, the program word 17 with the set bit 21 contained therein is checked for an active state. If such a state is identified, the further program word processing in the instruction word processing 10 is interrupted and the information part of this program word 17 is provided for the occupation of the read pointer register 18, write pointer register 19 and block length register 20 and is stored in these registers.

With these items of information newly written in the register contents, on the one hand, the modulo counting mode of the read pointer register 18 and of the write pointer register 19 is set with the content of the block length register 20. On the other hand, the respective start values of the read pointer 35 and write pointer 36 that are to be output during the next instruction word processing are preset with the contents of the read pointer register 18 and of the write pointer register 19.

During the subsequent instruction word processing in an instruction word memory 24 with a first or second instruction word memory page 6, 7 selected by the content of the page register 27, the output read pointer 35 specifies the current reading row number and the output write pointer 36 specifies the current writing row number of a block of instruction words which is to be processed and which is provided with the block length determined by the content of the block length register 20. The instruction word processing step can be executed in accordance with the program by means of (read/write/NOP no-operation) operations, connected with a required outputting operation into the instruction word outputting 11.

After an instruction word processing step, the read pointer register 18 and the write pointer register 19

are advanced by one and a further instruction word processing step can be effected.

If an interrupt signal 25 is triggered by the processor 13, an associated task with higher priority must be executed immediately. All of the previous register contents including those of the page register are buffer-stored and these registers are occupied anew during the interrupt operation. The instruction word processing associated with the interrupt is executed in an instruction word memory 24 provided therefor. After the end of the interrupt operation, all the buffer-stored register contents are restored and the instruction word processing can be continued in accordance with the program at the point interrupted by the interrupt signal 25.

If, during an instruction word processing, no new instruction word 15 is requested by the instruction word generation 10 for storage in the instruction word memory 24, the program word successor 16 from the sequence memory 9 and the instruction word 15 currently output by the instruction word memory 24 are read in by the prefetch unit 28 and the additional instruction word 29 is generated.

The latter is provided in a free row of the instruction word memory 24 or in the additional instruction word memory 30 and is immediately reloaded in the event of a request for a new instruction word 15 that is effected in the further course of the instruction word processing. The delay which [lacuna] during the provision of the new instruction word 15 in the instruction word generation 10, caused by the processing time arising there, is avoided in this way.

The functions required for processing in the corresponding functional units 10 of the processor 13

are triggered with storage of the instruction word 15 in the instruction word outputting 11.

Figure 2 reveals the generation unit 31, in which are arranged the block length register 20, the read pointer register 18 with the assigned read pointer up/down counter 32 and the write pointer register 19 with the assigned write pointer up/down counter 33.

The information part bus 26 is in each case present at the inputs of the block length register 20, of the read pointer register 18 and of the write pointer register 19, which bus provides the input values for the registers. The storage operation for the input values is effected by means of the set bit signal 34, which is likewise present at the inputs of the registers.

The output of the block length register 20 provides for [sic] the modulo count for the read pointer up/down counter 32 and the write pointer register 19. The start value of the read pointer 35 is provided at the output of the read pointer register 18 for inputting into the read pointer up/down counter 32. The output thereof applies the present value of the read pointer 35 to the input of the read pointer register 18. As a result, the read pointer 35 is provided via the second output of the read pointer register 18.

The start value of the write pointer 36 is provided at the output of the write pointer register 19 for inputting into the write pointer up/down counter 33. The output thereof applies the present value of the write pointer 36 to the input of the read pointer register 19 [sic]. As a result, the write pointer 36 is provided via the second output of the write pointer register 19.

Method and arrangement for instruction word generation in the driving of functional units in a processor

List of Reference Symbols

1	Program code
2	Primary instruction word
3	Instruction word memory
4	Instruction word part
6	First instruction word memory page
7	Second instruction word memory page
8	Program word generation
9	Sequence memory
10	Instruction word generation
11	Instruction word outputting
12	Functional unit
13	Processor
15	Instruction word
16	Program word successor
17	Program word
18	Read pointer register
19	Write pointer register
20	Block length register
21	Set bit
24	Instruction word memory
25	Interrupt signal
26	Information part bus
27	Page register
28	Prefetch unit
29	Additional instruction word
30	Additional instruction word memory
31	Generation unit
32	Read pointer up/down counter
33	Write pointer up/down counter
34	Set bit signal
35	Read pointer
36	Write pointer